

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 7, 2020

D. Trossen
Huawei
A. Rahman
C. Wang
InterDigital Communications, LLC
T. Eckert
Futurewei
February 4, 2020

Applicability of BIER Multicast Overlay for Adaptive Streaming Services
[draft-ietf-bier-multicast-http-response-03](#)

Abstract

HTTP Level multicast, using BIER, is described as a use case in the BIER Use Cases document. HTTP Level Multicast is used in today's video streaming and delivery services such as HLS, AR/VR etc., generally realized over IP Multicast as well as other use cases such as software update delivery. A realization of "HTTP Multicast" over "IP Multicast" is described for the video delivery use case. IP multicast is commonly used for IPTV services. DVB and BBF is also developing a reference architecture for IP Multicast service. A few problems with IP Multicast, such as waste of transmission bandwidth, increase in signaling when there are few users are described. Realization over BIER, through a BIER Multicast Overlay Layer, is described as an alternative. How BIER Multicast Overlay operation improves over IP Multicast, such as reduction in signaling, dynamic creation of multicast groups to reduce signaling and bandwidth wastage is described. We conclude with few next steps.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 7, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
1.1.	Reference Deployment	3
2.	Conventions used in this document	5
3.	Use Cases	5
3.1.	HTTP-based Steaming	6
3.2.	HTTP-based Software Updates	7
4.	Requirements	7
5.	Realization over IP Multicast	8
5.1.	Mapping to Requirements	9
5.2.	Problems	9
6.	Realization over BIER	10
6.1.	Description of a "BIER Multicast Overlay" to support HTTP Multicast	10
6.1.1.	BIER Multicast Overlay Components	11
6.1.2.	BIER Multicast Overlay Operations	11
6.2.	Achieving Multicast Responses	13
6.3.	BIER Multicast Overlay Traffic Management	14
7.	IANA Considerations	14
8.	Security Considerations	14
9.	Informative References	15
	Authors' Addresses	16

[1.](#) Introduction

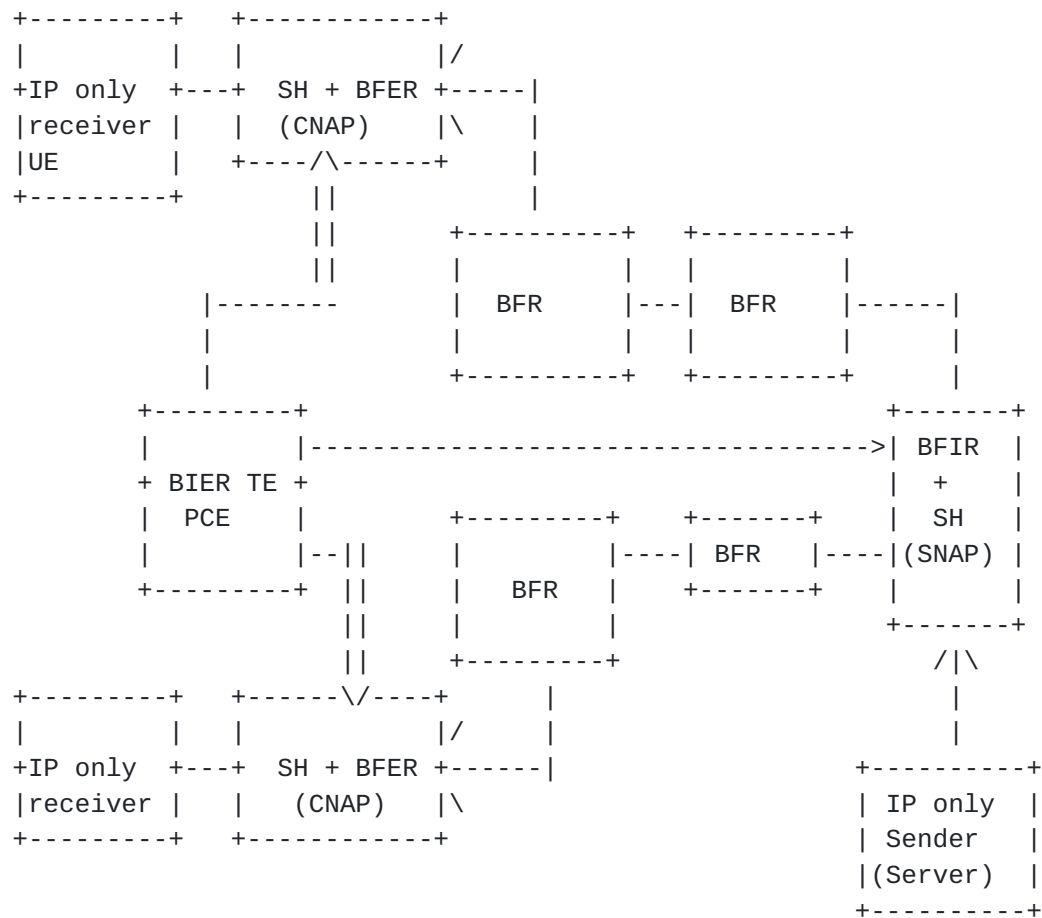
The BIER Use Cases document [[I-D.ietf-bier-use-cases](#)] describes an "HTTP Level Multicast" scenario, where HTTP Responses are carried over a BIER multicast infrastructure to multiple clients. Especially rate-adaptive HTTP solutions can benefit from the dynamic multicast group membership changes enabled by BIER. For this, the server-side Network Attachment Point (NAP), creates a list of outstanding client-side NAP requesting for the same HTTP resource. When the response is

available, the list of NAPs with outstanding client requests are converted into the BIER or BIER-TE bitstring and used to send the HTTP response.

In this draft, we describe use cases for such HTTP response multicast capability. Specifically for HTTP-based video streaming, we describe how this can be realized over IP Multicast and how the operation of the video delivery use case can be improved if realized over BIER. The realization over BIER is achieved through what is called "BIER Multicast overlay" layer, i.e., the methods by which the sending BIER router knows how to send other application packets. The requirements for BIER Multicast overlay layer is described in this document. It also describes the necessary functions that form the BIER multicast overlay and the operations that enable the desired "HTTP Level Multicast" behavior. One such operation is generating the PATH ID (represents the path between BFIR and BFER) based on named service relationship and translating it to appropriate BIER header. We describe a list of protocols needed for the realization of the individual operations.

1.1. Reference Deployment

Let us formulate the architecture of the BIER multicast overlay for the scenario outlined in [[I-D.ietf-bier-use-cases](#)]. This overlay is shown in Figure 1 below.



[SH : Service Handler, CNAP : Client Network Attachment Point]

[SNAP : Server Network Attachment Point]

[PCE : Path Computation Element]

Figure 1: Deployment over BIER

The multicast overlay is formed by the BFIR and BFER of the BIER layer and the additional Service Handler (SH) and Path Computation Element (PCE) elements shown in the figure. When interconnecting with a non-BIER enabled IP routed peering network, a special SH, such as Border Gateway may be used.

The Service Handler and BFER can be assumed to be collocated and can be viewed as Client Network Attachment Point (CNAP). Clients send and receive HTTP transactions through CNAP.

On the server side, the Service handling function can be part of the Server Network Attachment Point (SNAP). It includes the BFIR function and SH. SNAP is responsible for aggregating the relevant

HTTP Requests and sending one or more BIER Multicast HTTP response to multiple clients who requested the same content.

The SH function is assumed to be collocated with BFIR / BFER. The BFIR and BFER is assumed to be normal router boxes in the network. If the additional function of SH cannot be added to normal routers, then SH can be deployed as a separate function outside the routers. In such scenario an interface between SH and BFIR or BFER needs to be defined.

As part of the POINT/RIFE/FLAME EU Horizon 2020 projects, HTTP Level Multicast use case has been executed on SDN based and ICN based underlay network, as described in the [\[I-D.irtf-icnrg-deployment-guidelines\]](#).

"HTTP multicast" demonstrated benefits in HTTP-level streaming video delivery, when deployed on a POINT test bed with 80+ nodes. This draft [\[I-D.irtf-icnrg-deployment-guidelines\]](#) also describes protocol requirements to enable HTTP multicast to work on ICN underlay.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

3. Use Cases

With the extensive use of "web technology", "distributed services" and availability of heterogeneous network, HTTP has effectively transitioned into the common transport or session layer for E2E and multi-hop communication across the web that is also called Service signaling. Multi-hop when using a sequence of HTTP instance such as HTTP caches. The draft "On the use of HTTP as a Substrate" [\[I-D.ietf-httpbis-bcp56bis\]](#), describes how HTTP is commonly used among service instances to communicate with each other, thus abstracting the lower layer details to application developers.

For example, HTTP provides a common transport to support application layer streaming ([Section 3.1](#)) for not only conventional TV broadcasting, but also emerging Virtual Reality (VR) applications like VR-based tourist guide. HTTP can also be leveraged to support wide-area large-scale software updates ([Section 3.2](#)) such as for Vehicle-to-Everything (V2X) or Internet of vehicles use case. In the following, we present how such HTTP transport capability can be extended with multicast delivery for HTTP responses in certain use cases.

3.1. HTTP-based Steaming

Referring to the BIER Use Cases [[I-D.ietf-bier-use-cases](#)], multicast is used to scale out HTTP Live Streaming (HLS) to a large number of receivers that use HTTP. This is used today in solutions like DOCSIS hybrid streaming [[TR_IPMC_ABR](#)]. Multicast can speed up both live and high-demand VoD streaming. Adaptive Bit Rate IPMC [[TR_IPMC_ABR](#)] describes use of IP multicast towards the CMTS or a box beside it, where the content is converted to HTTP/TCP to stream to the receivers (e.g., homes). A server hosting the HLS content is shown as "NAP Server". The gateways acting as receivers for the multicast from the server are shown as "Client-NAP" (CNAP). Each CNAP can serve multiple clients.

Dynamic Adaptive Streaming (DASH) [[ISO_DASH](#)] over HTTP is another HTTP-based streaming approach. In DASH, each media is described by a Media Presentation Description (MPD) file, through which a DASH client (e.g. a media player) is instructed how to download, interpret and play the media. The media content is encoded into fragments or chunks at different bit rates. Both the MPD and media fragments are stored at a server. The DASH client first needs to retrieve the MPD file from the server; then it can start to retrieve media fragments encoded at different bits rates from the server. DASH players may use rate adaptation, i.e., switching the retrieval from one rate chunks to another rate. Usually this rate adaptation is utilizing delay measurements, resulting in TCP like behavior in terms of backoff in case of increasing delay. DASH has been designed to reuse most of existing Internet infrastructure and protocols and can run over different underlying transports including HTTP. For example, two major media service providers Netflix and Youtube use DASH over HTTP as their streaming technology.

HTTP request and response used in media streaming services like HLS and DASH over HTTP, use HTTP responses for delivery of content, i.e., each chunk is returned as an HTTP response to the requesting client. In such scenarios, where semi-synchronous access to the same resource occurs (such as watching prominent videos over Netflix or similar platforms or live TV over HTTP), traffic grows linearly with the number of viewers since the HTTP-based server will provide an HTTP response to each individual viewer. This poses a significant burden on operators in terms of costs and on users in terms of likely degradation of quality.

The use of HTTP-based streaming of video content is not limited to traditional TV broadcasting. Consider a virtual reality use case where several users are joining a VR session at the same time, e.g., centered around a joint event. Hence, due to the temporal correlation of the VR sessions, we can assume that multiple requests

are sent for the same content at any point, particularly when viewing angles of VR clients are similar or the same. Due to availability of virtual functions and cloud technology, the actual end point from where content is delivered may change.

3.2. HTTP-based Software Updates

Various new types of devices such as vehicles and robots are being connected to Internet. They could be physically located at or moving between different places and connect to Internet via different telecom operators. Software updates for these devices become important and introduce point-to-multipoint traffic from a software server to devices. Using V2X as an example, the software server could be a part of telecom operators or maintained by car manufacturers. In either case, the software server keeps vehicle software or firmware images, which will be transmitted to many vehicles across the global Internet, based on a pull or push model. HTTP is commonly used for those software updates to provide an E2E transport between the software server and each vehicle requesting software updates. As a result, the traffic from the software server to vehicles increases linearly with the number of connected vehicles since each vehicle will establish a HTTP connection with the software server.

4. Requirements

A realization for the "HTTP multicast" use case may have the following requirements:

- o MUST support multiple FQDN-based service endpoints to exist in the overlay to allow for utilizing several service endpoints for delivery and would therefore enable localization of content delivery.
- o MUST send FQDN-based service requests at the network level to a suitable FQDN-based service endpoint via policy-based selection of appropriate path information.
- o MUST allow for multicast delivery of HTTP response to same HTTP request URI.
- o MUST provide direct path mobility, where the path between the egress and ingress Service Routers(SR) can be determined as being optimal (e.g., shortest path or direct path to a selected instance), is needed to avoid the use of anchor points and further reduce service-level latency.

5. Realization over IP Multicast

We now discuss the realization of chunk-based delivery over IP multicast delivery methods. We focus our presentation here on the video streaming use case in [Section 3.1](#).

IPTV or Internet video distribution in CDNs, uses HTTP Level Multicast and realized over IP Multicast (IPMC). Many features of the IPTV service uses IPMC Group dependent state. Besides popular features like PIM, Mldp, in a variable bit rate encoded content source, content consumption also depends on group state.

DVB released reference architecture [[DVB_REF_ARCH](#)] for an end-to-end system to deliver linear content over IP networks in a scalable and standards-compliant manner. It focuses on delivering Adaptive Bit Rate unicast content over a IP multicast network.

A Multicast gateway is deployed in a CPE, Upstream Network Edge device or Terminal and provides multicast to unicast conversion facilities for several homes. All in-scope traffic on the access network between the Multicast Gateway (e.g. network edge device) and the Terminal or home gateway device is unicast. The individual media files are encapsulated into other protocols, so that they can be recovered as discrete files, when they exit the multicast pipe, which is terminated at Multicast Gateway. Interface "L" between Multicast server and Content playback supports fetching of all specified types of Content, Conditional request, Range request, Caching etc. BBF also started similar work in October 2016, called WT-399. This work is now coordinated with DVB. BBF focuses on developing the device management model.

Assume clients that are consuming the same content (such as a TV program) and that this content has for each block (typically segments worth 2 seconds of content) a set of outstanding requests from its clients. When IP Multicast is used in the domain, such as in aforementioned pre-existing solutions like in Cablelabs/DOCSIS [[TR_IPMC_ABR](#)], all possible blocks of the content have to be mapped to some IP multicast group, and the CNAP will need to know the mapping of block to groups. For example, a live stream may have 11 different bitrates available. In the most simple Block to IP multicast group mapping scheme, there could be 11 multicast groups, one for all the blocks of one bitrate (note that this is not necessarily done in deployments of this solution, but we consider it here for the purpose of explanation).

If the multicast domain and especially the links into the CNAP has enough bandwidth, this solution work well with IP multicast. As soon as there is at least one Client connected to a CNAP for one

particular content, the CNAP would join all 11 multicast groups for this content.

5.1. Mapping to Requirements

To realize "HTTP Level Multicast" over "IP Multicast", some additional functions needs to be supported in an intermediate (overlay) layer.

Support of mapping between FQDN based end points, Multicast Address.
Creating multicast group from FQDN based end points.

Control mechanism related to time when to start sending response as the multicast group is created. It is required that the source should not send response immediately to the Multicast address. Wait for some time to build the group sufficiently and then send response.

Support of IGMP signaling between User device, NAPs and Multicast Router.

5.2. Problems

If the number of clients on a CNAP for a particular program is large, the approach will work fairly well, because the likelihood that each of the 11 bitrates of a content is necessary for at least one Client is then fairly high.

When the number of receivers is not very large, IP multicast runs into two issues. If all the bitrates for the content are sent across the same group, then many of the bitrates may not be required and would have to be received unnecessarily and dropped by the CNAP. If each bitrate was sent on a different IP multicast group, the CNAP could dynamically join/leave each multicast group based on the known receivers, but that would create an extremely high and undesirable amount of IP multicast signaling protocol activity (PIM/IGMP) that is easily overloading the network

For efficiency reasons, the CNAP would need to dynamically join to only those bitrate streams where it does have outstanding requests, therefore achieving the best efficiency. This would mean in the worst case that a CNAP would need to send for each new block, aka.: every two second for every client one IGMP/PIM leave and one IGMP/PIM join towards the upstream router to get a block for an appropriate bitrate (or changed content) whenever bitrate or content on a client have changed. This high rate of control-plane signaling between CNAP and routers, and even between routers inside the multicast Domain is a major pain point and may easily prohibit deployment of these solutions because in many network devices, the performance of PIM/

IGMP is not scaled for continuous change in forwarding. Even worse, the limit may not simply be the CPU performance of the routers control plane, but a limitation in the number of changes in forwarding that the forwarding plane units (NPU/ASICs) can support.

6. Realization over BIER

6.1. Description of a "BIER Multicast Overlay" to support HTTP Multicast

The Service Handler (as in Figure 1) in BIER Multicast Overlay, process the FQDN in the service request. At the service level, e.g. HTTP service, the fixed relationship among consumer and providers may be abstracted using "Service Names", and the changing relationship at the Service execution endpoints can be managed at the "multicast overlay" level, handing out the exact locations where service request or response needs to be sent to BIER layer.

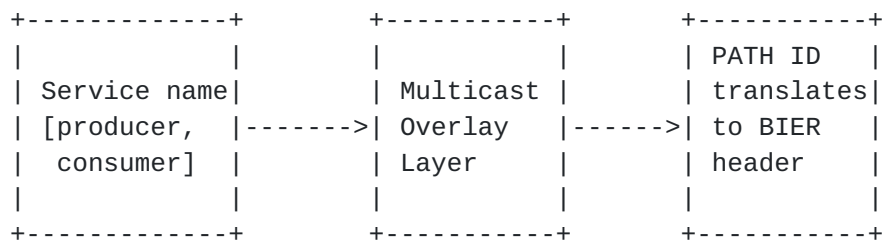


Figure 2: Service Name to Path ID Translation

We illustrate this using HTTP URI as service names. It should be noted, other identifiers can also be used as service name, such as an IP address. In the example illustration, other layers such as TCP, IP has been terminated at the egress point. Outside BIER domain we terminate TCP/IP session to extract the URI. The URI is processed by the "multicast overlay" layer to generate PATH IDENTIFIER, which is used as BIER header.

Path Identifier or PATH ID, is used in path-based approach, which utilizes path information provided by the source of the packet for forwarding said packet in the network. This is similar to segment routing albeit differing in the type of information provided for such source-based forwarding.

Once the BIER header is determined and added at the BFIR, the rest of the transport layers is assumed to be any underlay technology as supported by BIER. We assume TCP friendly transport, which can assure reliable delivery.

6.1.1. BIER Multicast Overlay Components

With reference to Figure 1, the following components are part of BIER Multicast Overlay Layer.

- o Service Handler (SH): The Service handler terminates transport level protocols, such as TCP, and extracts the URI. It processes the URI in order to determine the PATH ID by contacting the PCE for a suitable path resolution, which in turn is used to send the HTTP Request.
- o Optional PCE : Path Computation Element keeps track of all service execution end points through a registration process. SH interacts with the PCE to obtain PATH information by resolving the FQDN from the incoming URI at the ingress SH to a suitable PATH ID.
- o Interface functions to BFIR where the PATH ID is mapped to BIER header. An Interface to the BFER is likely not required because the BFER will only receive the traffic that they need and should be able to derive from the BIER payload which subset of its receivers need to get an HTTP encapsulated version of a particular reply.

6.1.2. BIER Multicast Overlay Operations

As shown in Figure 3, the "Multicast overlay function" includes a function called PCE (Path Computation Element function), which is responsible for selecting the correct multicast end point and possibly realizing path policy enforcement. The result of the selection is a BIER path identifier, which is delivered to the SH upon initial path computation request (or provided to the ingress router BFIR to be added as BIER header) (i.e., when sending a request to or response for a specific URL for the first time). The path identifier is utilized for any future request for a given URL-based request.

All service end points indicate availability to the PCE through a registration procedure, the PCE will instruct all SHs to invalidate previous path identifiers to the specific URL that might exist. This may result in an a renewed path computation request at the next service request forwarding. Through this, the newly registered service endpoint might be utilized if the policy-governed path computation selects said service instance. Otherwise, a previously resolved PATH ID for the URI determined at the ingress SH is being used instead, removing any resolution latency to an SH-local lookup of the PATH ID.

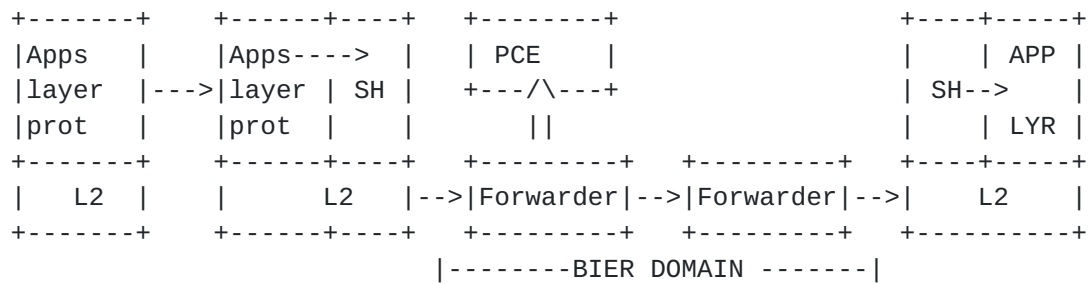


Figure 3: Protocol for Multicast Overlay Layer

In the diagram shown above, an HTTP request is sent by an IP-based device towards the FQDN of the server defined in the HTTP request.

At the client facing SH, the HTTP request is terminated at the TCP level at a local HTTP proxy. The server side SH at the egress terminates any transport protocol on the outgoing (server) side. These terminating functions are assumed to be part of the client/server SH. As a consequence, the SH obtains the destination "Service Name" from the received HTTP request.

If no local BIER forwarding information exists at the client side SH, the path computation entity (PCE) is consulted, which calculates a unicast path from the BFIR to which the client SH is connected to the BFER to which the server SH is connected. The PCE provides the forwarding information (Path ID) to the client SH, which in turn caches the result. The Client SH may forward the Path ID to BFIR, which creates the BIER header.

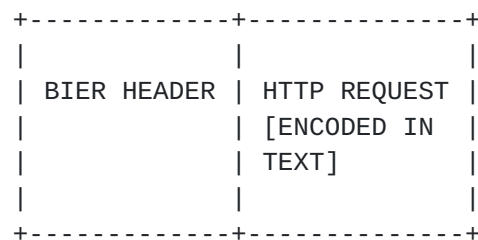


Figure 4: Encapsulation of Service Request

Ultimately, the "HTTP Request" encapsulated by BIER header, as shown in above diagram, is forwarded by the client SH towards the server-facing SH via the local BFIR. We assume a (TCP-friendly) transport protocol being used for the transmission between client and server SH. The possibility of sending one HTTP response to several CNAPs makes this a reliable multicast transport protocol. The exact nature

of this transport protocol is left for further studies. A suitable transport or Layer 2 encapsulation, as supported by BIER layer, is added to the above payload.



Figure 5: Transport Encapsulation of BIER payload

Upon arrival of an HTTP request at the server SH, it forwards the HTTP request as a well-formed HTTP request locally to the server, awaiting an HTTP response for the reverse direction.

If no BIER forwarding information exists for the reverse direction towards the requesting client SH, this information is requested from the PCE, similar to the operation in forward direction.

6.2. Achieving Multicast Responses

Upon arrival of any further client SH request at the server SH to an HTTP request whose response is still outstanding, the client SR is added to an internal request table. Optionally, the request is suppressed from being sent to the server.

Upon arrival of an HTTP response at the server SH, the server SH consults its internal request table for any outstanding HTTP requests to the same request. The server SH retrieves the stored BIER forwarding information for the reverse direction for all outstanding HTTP requests and determines the path information to all client SHs through a binary OR over all BIER forwarding identifiers with the same SI field. This newly formed joint BIER multicast response identifier is used to send the HTTP response across the network.

BIER makes the solution scalable. Instead of IP multicast with IGMP/PIM, BIER is being used between Server NAP (SNAP) and CNAP, the SNAP simply coalesces the forwarded HTTP requests from the CNAP, and determines for every requested block the set of CNAPs requesting it. A set of CNAPs corresponds to a set of bits in the BIER-bitstring, one bit per CNAP. The SNAP then sends the block into BIER with the appropriate bitstring set.

This completely eliminates any dynamic multicast signaling between CNAP and SNAP. It also avoids sending of any unnecessary data block, which in the IP multicast solution is pretty much unavoidable.

Furthermore, using the approach with BIER, the SNAP can also easily control how long to delay sending of blocks. For example, it may wait for some percentage of the time of a block (e.g, 50% = 1 second), therefore ensuring that it is coalescing as many requests into one BIER multicast answer as possible.

6.3. BIER Multicast Overlay Traffic Management

BIER-TE (BIER Traffic Engineering [[I-D.ietf-bier-te-arch](#)]) forwards and replicates packets like BIER based on a BitString in the packet header. Where BIER forwards and replicates its packets on shortest paths towards BFER, BIER-TE allows (and requires) to also use bits in the bitstring to indicate the paths in the BIER domain across which the BIER-TE packets are to be sent. This is done to support Traffic Engineering for BIER packets via explicit hop-by-hop and/or loose hop forwarding of BIER-TE packets. A BIER-TE controller calculates explicit paths for this packet forwarding.

The Multicast Flow Overlay operates as in BIER. Instead of interacting with the BIER layer, it interacts with the BIER-TE Controller.

In this draft, "Name-based" service forwarding over BIER, is described to handle changes in service execution end points and manage adhoc relationship in a multicast group. BIER-TE is another way of doing this, while integrated with BIER architecture. The PCE function described earlier in the BIER Multicast Overlay, may become part of BIER-TE Controller. The SH function in the CNAP and SNAP communicates with BIER TE controller. SH sends the service name to the controller, which process the request using the PCE function and returns the "bitstring" to be used as BIER header for delivery of the HTTP response to multiple clients.

7. IANA Considerations

This document requests no IANA actions.

8. Security Considerations

The operations in [Section 6](#) consider the forwarding of HTTP packets between ingress and egress points based on information derived from the HTTP request. The support for HTTPS is foreseen to ensure suitable encryption capability of such exchanges. For this to happen, we expect certificate sharing agreements to exist between the

content provider and the BIER overlay provider, ensuring the extraction of the suitable request parameters while allowing for the re-encryption of the content for an encrypted delivery over the BIER network. Since we liken the relationship between content and BIER overlay provider to that between content and CDN provider, the existence of certificate sharing agreements is similar to the practice used for CDNs.

9. Informative References

[DVB_REF_ARCH]

DVB, "Adaptive media streaming over IP multicast", DVB Document A176, March 2018, <https://www.dvb.org/resources/public/standards/a176_adaptive_media_streaming_over_ip_multicast_2018-02-16_draft_bluebook.pdf>.

[I-D.ietf-bier-te-arch]

Eckert, T., Cauchie, G., Braun, W., and M. Menth, "Traffic Engineering for Bit Index Explicit Replication (BIER-TE)", [draft-ietf-bier-te-arch-00](#) (work in progress), January 2018.

[I-D.ietf-bier-use-cases]

Kumar, N., Asati, R., Chen, M., Xu, X., Dolganow, A., Przygienda, T., Gulko, A., Robinson, D., Arya, V., and C. Bestler, "BIER Use Cases", [draft-ietf-bier-use-cases-10](#) (work in progress), August 2019.

[I-D.ietf-httpbis-bcp56bis]

Nottingham, M., "On the use of HTTP as a Substrate", [draft-ietf-httpbis-bcp56bis-05](#) (work in progress), May 2018.

[I-D.irtf-icnrg-deployment-guidelines]

Rahman, A., Trossen, D., Kutscher, D., and R. Ravindran, "Deployment Considerations for Information-Centric Networking (ICN)", [draft-irtf-icnrg-deployment-guidelines-07](#) (work in progress), September 2019.

[ISO_DASH]

ISO, "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats", ISO/IEC 23009-1:2014, May 2014, <<http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[TR_IPMC_ABR]
CableLabs, "IP Multicast Adaptive Bit Rate Architecture Technical Report", OC-TR-IP-MULTI-ARCH-V01-141112 C01, October 2016, <<https://community.cablelabs.com/wiki/plugins/servlet/cablelabs/alfresco/download?id=51b3c11a-3ba4-40ab-b234-42700e0d4669;1.0>>.

Authors' Addresses

Dirk Trossen
Huawei Technologies Duesseldorf GmbH
205 Hansallee
Duesseldorf 40549
Germany

Email: dirk.trossen@huawei.com
URI: <http://huawei-dialog.de/>

Akbar Rahman
InterDigital Communications, LLC
1000 Sherbrooke Street West
Montreal H3A 3G4
Canada

Email: Akbar.Rahman@InterDigital.com

Chonggang Wang
InterDigital Communications, LLC
1001 E Hector St
Conshohocken 19428
USA

Email: Chonggang.Wang@InterDigital.com

Toerless Eckert
Futurewei Technologies Inc.
2330 Central Expy
Santa Clara 95050
USA

Email: tte+ietf@cs.fau.de